METHOD FOR UTILIZING RESOURCE CHARACTERIZATIONS TO OPTIMIZE PERFORMANCE IN AN ELECTRONIC DEVICE

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to, and claims priority in, co-pending U.S. Provisional Patent Application Serial No. 60/161,206, entitled "Method For Implementing Scheduling Mechanisms By Utilizing Resource Characterizations," filed on October 21, 1999, and to co-pending U.S. Provisional Patent Application Serial No. 06/160,991, entitled “Method For Quantifying Available System Resources Associated With A Hardware Component,” filed on October 21, 1999. All of these related applications are commonly assigned, and are hereby incorporated by reference.

BACKGROUND SECTION

1. Field of the Invention

This invention relates generally to techniques for implementing electronic devices, and relates more particularly to a method for utilizing resource characterizations to optimize performance in an electronic device.

2. Description of the Background Art

Implementing effective methods for utilizing device resources is a significant consideration for designers and manufacturers of contemporary electronic devices. However, effectively utilizing device resources may create substantial challenges for designers of electronic devices. For example, enhanced demands for increased device functionality and performance may require more system processing power, bus bandwidth, and require additional hardware resources. An increase in processing or hardware requirements may also result in a corresponding detrimental economic impact due to increased production costs and operational inefficiencies.

An electronic device in an electronic network may advantageously communicate with other electronic devices in the network to share resources to thereby substantially increase the capabilities and versatility of individual devices in the electronic network. For example, an electronic network may be implemented in a home environment to enable flexible and beneficial sharing of data and device resources between various consumer electronic devices, such as personal computers, digital video disc (DVD) devices, digital set-top boxes for digital broadcasting, enhanced television sets, and audio reproduction systems.

Network size is also a factor that affects the management of resources in an electronic network. Communications in an electronic network typically become more complex as the number of individual devices or nodes increases. A local software module on the local device may need to communicate with various remote software elements on remote devices across the electronic network. However, successfully managing resources of a substantial number of electronic devices across a network may provide significant benefits to a system user.

Furthermore, enhanced device capability to perform various advanced processes may provide additional benefits to a system user, but may also place increased demands on the control and management of an electronic device. For example, an enhanced electronic device that effectively accesses, processes, and displays digital television programming may benefit from efficient use of resources because of the large amount and complexity of the digital data involved.

Due to growing demands on system resources and substantially increasing data magnitudes, it is apparent that developing new and effective methods for managing resources is a matter of importance for the related electronic technologies. Therefore, for all the foregoing reasons, implementing effective methods for utilizing resources remains a significant consideration for designers, manufacturers, and users of contemporary electronic devices.

SUMMARY

In accordance with the present invention, a method is disclosed for effectively utilizing resource characterizations to optimize performance in an electronic device. In one embodiment of the present invention, initially, device software preferably generates an isochronous process request to a cantaloupe manager that functions as a resource allocation manager for the electronic device. In response, the cantaloupe manager preferably accesses resource usages or resource requirements that are listed in one or more resource characterizations known as "cantaloupes". The resource usages in an accessed cantaloupe preferably correspond to the foregoing isochronous process that was initially requested by the device software.

The cantaloupe manager then preferably may compare the resource usages from the cantaloupe(s) with currently-available resources of the electronic device. In certain embodiments, the cantaloupe manager may sequentially compare each individual resource usage from the cantaloupe with a corresponding current available resource of the electronic device.

If sufficient available resources are currently present for optimal performance of the requested isochronous process, then the cantaloupe manager preferably authorizes the device software to instantiate the requested process through a picokernel module. However, if sufficient currently-available resources are not present for optimal execution of the requested isochronous process, then the cantaloupe manager preferably generates a request-fail signal to the device software to thereby deny the request to instantiate the isochronous process.

In this manner, the present invention advantageously pre-allocates sufficient guaranteed resources for a given isochronous process, prior to instantiation, to thereby guarantee successful and deterministic performance of the requested isochronous process. The present invention therefore provides an effective method for utilizing resource characterizations to optimize performance in an electronic device.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram for one embodiment of an electronic network, in accordance with the present invention;

FIG. 2 is a block diagram for one embodiment of an exemplary device of FIG. 1, in accordance with the present invention;

FIG. 3 is a diagram for one embodiment of the memory of FIG. 2, in accordance with the present invention;

FIG. 4 is a block diagram that illustrates non-optimal performance in an electronic device;

FIG. 5 is a block diagram for one embodiment of a cantaloupe, in accordance with the present invention;

FIG. 6 is a block diagram that illustrates a resource allocation procedure, in accordance with one embodiment of the present invention; and

FIG. 7 is a flowchart of method steps for performing a resource allocation procedure, in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION

The present invention relates to an improvement in electronic devices. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment will be readily apparent to those skilled in the art and the generic principles herein may be applied to other embodiments. Thus, the present invention is not intended to be limited to the embodiment shown, but is to be accorded the widest scope consistent with the principles and features described herein.

A method for providing optimal performance in an electronic device is herein disclosed that comprises at least one resource characterization which includes resource requirements for executing a requested process. An allocation manager may then compare the resource requirements for the requested process and the currently-available device resources. The allocation manager may advantageously authorize or deny the requested process depending upon whether the currently-available resources are sufficient to adequately service the requested process.

Referring now to FIG. 1, a block diagram for one embodiment of an electronic network 110 is shown, in accordance with the present invention. In the FIG. 1 embodiment, network 110 preferably comprises, but is not limited to, a number of electronic devices 112 (device A 112(a), device B 112(b), root device 114, device C 112(c), device D 112(d), and device E 112(e)). In alternate embodiments, electronic network 110 may readily be configured to include various other devices or components that function in addition to, or instead of, those discussed in conjunction with the FIG. 1 embodiment. In alternate embodiments, network 110 may readily be connected and configured in any other appropriate and suitable manner.

In the FIG. 1 embodiment, devices 112 of network 110 may be implemented as any type of electronic device, including, but not limited to, personal computers, printers, digital video disc devices, television sets, audio systems, video cassette recorders, and set-top boxes for digital broadcasting. In the FIG. 1 embodiment, devices 112 preferably communicate with one another using a network bus 132. Network bus 132 preferably includes path 132(a), path 132(b), path 132(c), path 132(d), and path 132(e). For example, in one embodiment, device B 112(b) is coupled to device A 112(a) via path 132(a), and to root device 114 via path 132(b). Similarly, root device 114 is coupled to device C 112(c) via path 132(c), and to device D 112(d) via path 132(d). In addition, device D 112(d) is coupled to device E 112(e) via path 132(e). In the FIG. 1 embodiment, network bus 132 is preferably implemented using an IEEE Std 1394 Standard for a High Performance Serial Bus, which is hereby incorporated by reference. However, in alternate embodiments, network 110 may readily communicate and function using various other network interconnectivity methodologies which are equally within the scope of the present invention.

In the FIG. 1 embodiment, each device in electronic network 110 may preferably communicate with any other device within network 110. For example, device E 112(e) may communicate with device B 112(b) by transmitting transfer data via cable 132(e) to device D 112(d), which then may transmit the transfer data via cable 132(d) to root device 114. In response, root device 114 then may transmit the transfer data to device B 112(b) via cable 132(b). In the FIG. 1 embodiment, root device 114 preferably provides a master cycle start signal to synchronize isochronous processes for devices 112 in network 110. In other embodiments of network 110, any one of the network devices 112 may be designated as the root device or cycle master.

Referring now to FIG. 2, a block diagram for one embodiment of an exemplary device 112 from network 110 is shown, in accordance with the present invention. Device 112 preferably includes, but is not limited to, a

processor 212, an input/output (I/O) interface 214, a memory 216, a device bus 226, and a bus interface 220. Processor 212, I/O interface 214, memory 216 and bus interface 220 preferably are each coupled to, and communicate via common device bus 226.

In the FIG. 2 embodiment, processor 212 may be implemented as any appropriate multipurpose microprocessor device. Memory 216 may be implemented as one or more appropriate storage devices, including, but not limited to, read-only memory, random-access memory, and various types of non-volatile memory, such as floppy disc devices or hard disc devices. I/O interface 214 preferably may provide an interface for communications with various compatible sources and/or destinations.

In accordance with the present invention, bus interface 220 preferably provides an interface between device 112 and network 110. In the FIG. 2 embodiment, bus interface 220 preferably communicates with other devices 112 on network 110 via network bus 132. Bus interface 220 also preferably communicates with processor 212, I/O device 214, and memory 216 via a common device bus 226.

In the FIG. 2 embodiment, device 112 preferably includes the capability to perform various tasks that involve isochronous data and isochronous processes. Isochronous data typically includes information that is time-sensitive, and therefore requires deterministic operations to guarantee delivery and processing of the isochronous data in a timely manner. For example, video data that is intended for immediate display must arrive at the appropriate destination in a time-synchronized manner in order to prevent jitter or breakup of the corresponding image during display. To achieve this goal, device 112 preferably performs isochronous and other types of processing in segments of time called "cycles". Isochronous processes are synchronized with a cycle clock or submultiples thereof. Processes that are synchronized loosely with a sub-multiple of the cycle clock are called plesiochronous processes. *

extra

Scheduling of isochronous processes typically requires a finite time period that is sometimes referred to as "overhead". As the cycle time period is

reduced, the overhead becomes a more significant factor because of the reduced amount of time remaining to perform the actual isochronous transfer. In the FIG. 2 embodiment, the cycle time period may be in the proximity of 125 microseconds, with a cycle frequency of approximately eight kilohertz.

Referring now to FIG. 3, a diagram for one embodiment of the FIG. 2 memory 216 is shown, in accordance with the present invention. In the FIG. 3 embodiment, memory 216 preferably includes, but is not limited to, device software 312, picokernel 314, cantaloupe manager 316, cantaloupe(s) 318, and resource values 320. In alternate embodiments, memory 216 may readily include various other components in addition to, or instead of, the components that are discussed in conjunction with the FIG. 3 embodiment.

In the FIG. 3 embodiment, device software 312 includes software instructions that are preferably executed by processor 212 for performing various functions and operations by device 112. The particular nature and functionality of device software 312 preferably varies depending upon factors such as the type and purpose of the corresponding host device 112.

In the FIG. 3 embodiment, picokernel 312 preferably controls and coordinates the scheduling of isochronous processes by utilizing an optimized process representation to reduce the cost or overhead of scheduling to a minimum. Cantaloupe manager 316 preferably includes an allocation manager that may utilize information from cantaloupe(s) 318 to determine whether a particular isochronous process may be instantiated on behalf of another entity, such as device software 312. Cantaloupe(s) 318 preferably comprise a resource characterization that includes one or more characterizations of hardware and/or software resources necessary to meet performance criteria for a particular isochronous process. Cantaloupe(s) 318 are further discussed below in conjunction with FIGS. 5 through 7.

Resource values 320 preferably include any relevant information regarding current resource availability and allocations in device 112. For example, in the FIG. 3 embodiment, resource values 320 may include one or

more available resource value(s), one or more allocated resource value(s), and one or more total device resource value(s) for device 112. In the FIG. 3 embodiment, prior to allocation of any resources, the available resource value(s) may initially be set to a value that is less than 100% of total device resource values (such as 75%) to thereby reserve resources necessary for non-isochronous processes or system tasks.

Referring now to FIG. 4, a block diagram that illustrates non-optimal performance in an exemplary electronic device 412 is shown. In the FIG. 4 embodiment, device 412 preferably instantiates a process A 424 that is performed using software 416 and hardware 420. For example, process A 424 may consume 75% of the total resources available on device 412 to decode and display video programming. In the FIG. 4 embodiment, device 412 then preferably instantiates a process B 430 that is performed using software 416 and hardware 420. For example, process B 430 may require 35% of the total resources available on device 412 to perform a speech recognition function.

Simultaneously executing process A 424 (using 75% of total available resources) and process B 430 (using 35% of total available resources) requires more than 100% of the total resources available from device 412 (75% + 35% = 110%). Therefore, insufficient resources are available for simultaneously executing process A 424 and process B 430. Device 412 may attempt to simultaneously execute process A 424 and process B 430 by reducing the amount of resources provided to one or both of the simultaneously-executing processes.

Executing a particular process without providing sufficient resources may result in non-optimal performance or "graceful degradation". For example, if process A 424 lacks sufficient resources for successful performance, such graceful degradation may include the disruption of video information that is being displayed to a system viewer. In many circumstances, such degradation of device performance is not desirable or acceptable as a performance model for many electronic devices.

Referring now to FIG. 5, a diagram for one embodiment of a cantaloupe 318 is shown, in accordance with the present invention. In the FIG. 5 embodiment, cantaloupe 318 preferably includes a listing for a resource 1 (512(a)) through a listing for a resource N (512(c)). In the FIG. 5 embodiment, resources 512 may include any appropriate aspects of devices 112 or network 110 (FIG. 1). For example, resource 512 may correspond to the bandwidth for a particular bus, such as device bus 226 or network bus 132. Similarly, resource 512 may correspond to the processing capacity for a central processing device, such as processor 212 (FIG. 2), or to the capacity of a memory device, such as memory 216. In alternate embodiments of the present invention, cantaloupe 318 may readily be implemented to include various other configurations, and may also include various items and components that are different from those discussed in conjunction with the FIG. 5 embodiment.

In accordance with the present invention, cantaloupe 318 is preferably associated with a particular time-sensitive isochronous or plesiochronous process on network 110. Prior to instantiating the foregoing process, cantaloupe manager 316 may then advantageously reference cantaloupe 318 to determine the individual and total resources necessary for the associated process. Determining in advance whether sufficient system resources are available for successful operation of a given process ensures that the associated isochronous process is guaranteed sufficient resources for timely and deterministic performance. Providing sufficient resources becomes more significant as the cycle duration decreases and the cycle frequency increases. In one embodiment of the present invention, the cycle period may preferably be in the proximity of 125 microseconds, with a rate of eight kilohertz.

In the FIG. 5 embodiment, cantaloupe 318 preferably also includes a listing for a resource 1 usage (514(a)) through a listing for a resource N usage (514(c)). In the FIG. 5 embodiment, each of resource usages 514 preferably corresponds with a given resource 512 to characterize the amount of the

given resource 512 required by the isochronous process associated with cantaloupe 318.

For example, if a given resource 512 is the bandwidth for a particular bus, then the corresponding resource usage 514 may be expressed in bandwidth units utilized by a process. Similarly, if a given resource 512 is the processing capacity for a central processing device, then corresponding resource usage 514 may be expressed in CPU units, such as machine-instructions per second (MIPS). In various embodiments of the present invention, resource usages 514 may be implemented in any appropriate and compatible format for use by network 110. In one embodiment, cantaloupe 318 may be implemented using only resource usages 514, with the corresponding resources 512 indirectly implied and understood during instantiation of process on network 110. In certain embodiments, cantaloupe 318 may also be utilized to characterize other resources, such as the total system resources, or the current available resources of device 112.

Therefore, cantaloupe 318 preferably includes an at least two-dimensional array of descriptive parameters. The first parameter preferably may be the type of resource being characterized, and the second parameter is the amount of required resource usage. A cantaloupe 318 may thus serve as a common descriptor to couple hardware and software scheduling mechanisms by describing resource requirements. For example, resource usage may be characterized and described as a ratio of the amount of usage per a given time period (including process scheduling overhead).

Referring now to FIG. 6, a block diagram illustrating the use of cantaloupe 318 is shown, in accordance with one embodiment of the present invention. In alternate embodiments, cantaloupe 318 may readily be utilized in various other manners and configurations, in accordance with the present invention.

In the FIG. 6 embodiment, device software 312 initially generates an isochronous process request to cantaloupe manager 316 via path 618. In response, cantaloupe manager 316 preferably accesses the resource usages

514 in cantaloupe 318 via path 614. In the FIG. 6 embodiment, cantaloupe 318 preferably corresponds to the foregoing isochronous process that was initially requested by device software 312.

Cantaloupe manager 316 then preferably compares the resource usages 514 from cantaloupe 318 with available resources 320 for the requested process via path 616. In the FIG. 6 embodiment, cantaloupe manager 316 may sequentially compare each individual resource usage 514 from cantaloupe 318 with a corresponding currently-available associated system resource 320.

If sufficient additional current resources 512 are available for the requested isochronous process, then cantaloupe manager 316 preferably authorizes device software 312 to schedule and instantiate the requested process through picokernel 314. However, if sufficient current resources 512 are not available for the requested isochronous process, then cantaloupe manager 316 preferably generates a request fail signal to device software 312 to deny authorization of the requested isochronous process. In this manner, the present invention advantageously pre-allocates sufficient resources for a given isochronous process, prior to instantiation, to thereby guarantee successful and deterministic performance of the isochronous process.

Referring now to FIG. 7, a flowchart of method steps for performing a resource allocation procedure is shown, in accordance with one embodiment of the present invention. In alternate embodiments, the FIG. 7 resource allocation procedure may readily be performed in various other manners and sequences, in accordance with the present invention.

In the FIG. 7 embodiment, initially, in step 712, an entity (such as device software 312) preferably generates a request for instantiation of an isochronous process. In response, in step 716, cantaloupe manager 316 preferably examines one or more cantaloupes 318 that correspond to the requested isochronous process.

In step 720, cantaloupe manager 316 preferably determines whether sufficient resources are currently available for performing the requested

isochronous process. In accordance with the present invention, cantaloupe manager 316 may utilize any appropriate technique to determine whether sufficient resources are available for performing the requested process.

In the FIG. 7 embodiment, cantaloupe manager 316 preferably maintains one or more available resource values in resource values 320 of memory 216 (FIG. 3) to quantitatively represent any currently unallocated resources. For example, available resource values may be expressed as percentages of total system resources, or as a finite resource amount. Cantaloupe manager 316 may then compare the resources required for the requested process (obtained from cantaloupe(s) 318) and the currently-available resource value(s) to determine whether sufficient unallocated resources are available for utilization by the requested process.

In step 720, if cantaloupe manager 316 determines that sufficient resources are not available for performing the requested process, then, in step 724, cantaloupe manager 316 preferably denies the request for instantiation of the process, and the FIG. 7 method terminates. However, if cantaloupe manager 316 determines that sufficient resources are available for performing the requested process, then, in step 728, cantaloupe manager 316 preferably allocates the required resources, and thereby allows granting of the request for instantiation of the isochronous process.

In step 732, cantaloupe manager 316 preferably updates the available resource value(s) in memory 216 to exclude the resources that were allocated in foregoing step 728 to service the requested isochronous process. For example, if the requested process requires twenty-five percent of the managed resources, then, cantaloupe manager 316 preferably may decrease the available resource value in memory 216 by twenty-five percent. Finally, in step 736, picokernel 314 of device 112 preferably may instantiate and execute the requested isochronous process. The resources that are allocated for the isochronous process are therefore guaranteed to be available, and the isochronous process is thus assured of successful execution without degraded or non-optimal performance. In accordance with the present

invention, the FIG. 7 process may readily be utilized to evaluate a series of requested isochronous processes.

The invention has been explained above with reference to a preferred embodiment. Other embodiments will be apparent to those skilled in the art in light of this disclosure. For example, the present invention may readily be implemented using configurations and techniques other than those described in the preferred embodiment above. Additionally, the present invention may effectively be used in conjunction with systems other than the one described above as the preferred embodiment. Therefore, these and other variations upon the preferred embodiments are intended to be covered by the present invention, which is limited only by the appended claims.